Session 507

QuickTime for Java

Jon Summers and Michael Hopkins
QTJava Engineering

QuickTime and Java

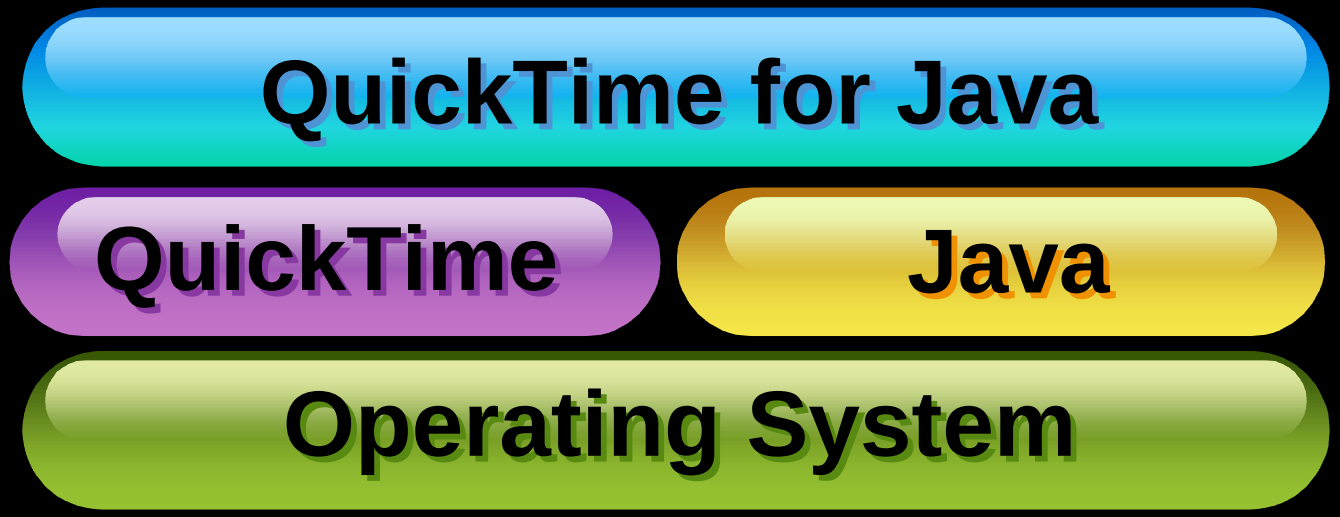
- QuickTime
  - Mature, cross-platform, flexible media architecture
  - Rich set of services supporting industry-standard media
    - Dynamic media, still images, interactive media
- Java
  - Modern object-oriented interpreted language
  - Cross-platform deployment of applications and applets

QuickTime for Java

- QuickTime for Java is a cross-platform Java class library accessing the native QuickTime C libraries
  - Core layer object model that represents QuickTime "C" API as a set of Java classes in logical packages
  - Application framework of abstracted media services to simplify the integration of QuickTime and Java
  - Identical API across supported platforms and operating systems

QTJava Architecture

QuickTime for Java relies on core services from QuickTime, Java and the Operating System



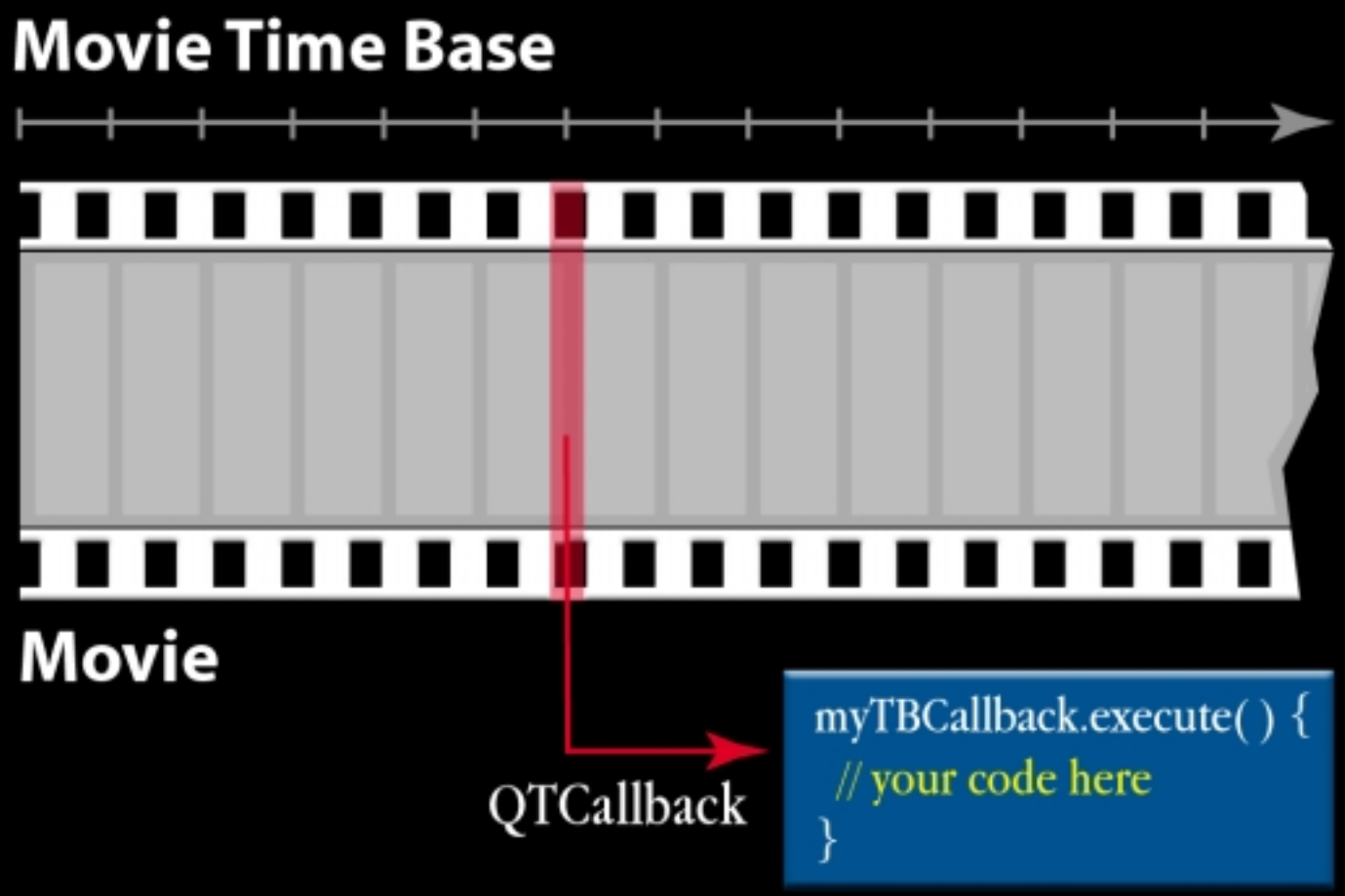
QTJava 5 and Java

- Supports the following platforms:
  - Mac OS X with J2SE™
  - Mac OS with MRJ 2.1 and above (JDK 1.1.8)
  - Windows 95 and NT families with Sun-compliant JRE 1.1 and Java 2
- Specific enhancements for Java 2
  - Improvements for Java 2 Security Model
  - Changes for GC issues with Hotspot compiler

QTJava in QuickTime 5

- QuickTime for Java 5 release
  - Supports new features in QuickTime 5 API
    - XMLParser package for QuickTime XML API
    - Presentation package for Broadcasting API
  - Full coverage of QuickTime 4 API
  - Additional functionality and enhanced services

Time Based Callbacks



TimeBase Callbacks

- Callbacks execute on specific time-based events
  - At a certain time, at intervals, at extremes, and on rate change
- When you use them?
  - Receive notification of time based events
  - Start/stop (rate change), begin/end, periodic, time, and time jump
- Callbacks re-worked for QTJava 5
  - High priority native threads for efficiency
  - Implicit idling eliminates need for a tasking thread

Demo

Using Movie Callbacks

Jon Summers

TimeBase Callbacks

- Some timely advice
  - TimeBase callbacks require explicit cleanup
  - On exit must use cancelAndCleanup()
  - Use cancel() to stop or reschedule the callback
- Application Framework time services
  - Timer class/Ticklish interface
  - Time Slaving techniques

All your TimeBase are belong to us

New QTVRInstance Methods—Interaction Properties

- Panorama MouseClick Properties
  - MouseClickHysteresis
  - MouseClickTimeout
- Speed of angle change Properties
  - PanTiltSpeed
  - MouseMotionScale
  - ZoomSpeed
- VRObject Rotation/Translation Properties
  - TranslateOnMouseDown
  - NudgeMode

Demo

QTVR Interaction Routines

QTVR Interaction Routines

- Using interaction properties

```
QTVRInstance vr = new QTVRInstance (mov.getQTVRTrack(1), mc);

vr.setMouseOverProc (new myOverHotSpot(), 0);

class  myOverHotSpot implements QTVRMouseOverHotSpot {
    public boolean execute (QTVRInstance vr, int hsID, int action) {
        if  (action == kQTVRHotSpotEnter)
            vr.setInteractionPanTiltSpeed(myChooseSpeed(hsID));
        else if (action == kQTVRHotSpotLeave)
            vr.setInteractionPanTiltSpeed (5);  // normal speed
        return 0;
    }
}
```

Improved Support for Sound Manager

Michael Hopkins
QTJava Engineering

Sound Manager Support in QuickTime for Java 5

- Additional support for Sound Manager
  - Asynchronous recording and playback
  - SndChannel EQ, treble and bass
  - Level Metering
- Classic SoundManager support (not CoreAudio)

Sound Manager Improvements

- ExtScheduledSoundHeader class
  - For sending buffered sound commands
- SCStatus class
  - Get status of sound channel
- SICompletion
  - For asynchronous recording
- Additional constants in SoundConstants

Sound Manager Improvements (Cont.)

- SndChannel class enhanced
  - **doCommand(…)**
  - **doImmediate(…)**
  - callbackCmd
  - information accessors
    - **getLevelMeterOnOff(…)**
    - **setLevelMeterOnOff(…)**

Recording a Sampled Sound

- Preparing a sound channel
- Create a new Sound Input Device

```
new SPBDevice (null, permFlags);
```

- Allocate a new SoundHandle

```
new SndHandle (numChannels,
    sampleRate, sampleSize,
    compressionType, bassFreq);
```

Recording a Sampled Sound (Cont.)

- Creating the Sound Input Parameter Block

```
recorder = new SPB (sndDevice, 0, sndBufferSize,
    sndHndl.getSoundData());
```

- Setting the completion Proc

```
recorder.setCompletionProc (new SICompletion ( ) {
    public void execute (SPB paramBlock) { … }
});
```

- Recording the sound

```
recorder.record(true);
```

Playing a Sound Asynchronously

- Playing the sound data

```
SndCommand cmd =  new SndCommand (bufferCmd);
cmd.setBuffer (buffer);          // extended sound header
sc.doCommand (cmd);
```

- Scheduling a callback to be executed when callBackCmd is queued

```
SndChannel sndChan1 = new SndChannel (
    new SoundCallBack () {
        public void execute (SndChannel sc) {…}
});
```

Some Sound Advice

- Dispose of sound channels when done

  sc.disposeQTObject();

  - This can be done within the callback
- Remove completion proc from SPB

  spb.removeCompletionProc();
- Garbage Collection issues
  - Local object variables may go out of scope
  - Dispose objects that are used in callback, then dispose of callback

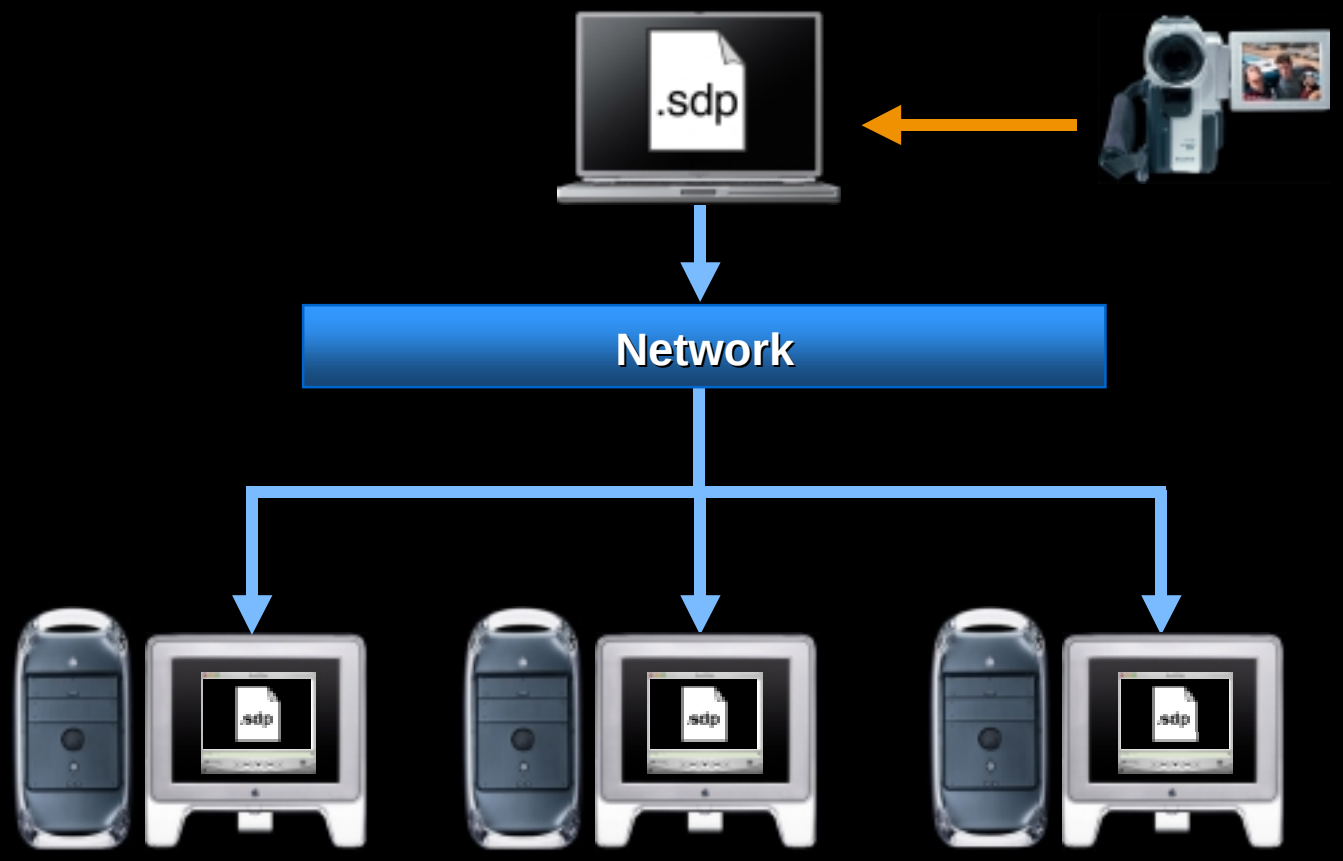
Using the Presentation API

Michael Hopkins
QTJava Engineering

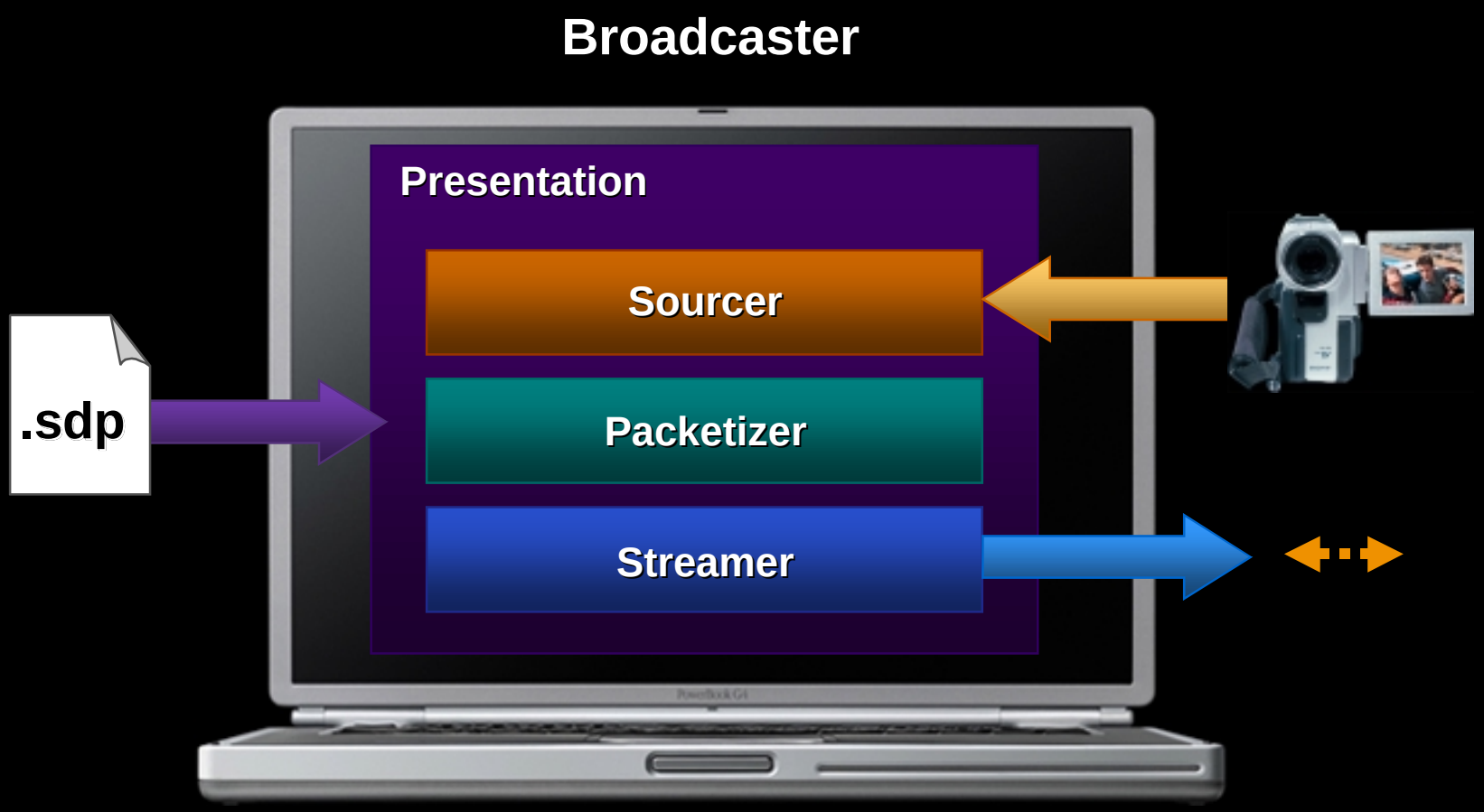
Using the Presentation API

- New API with QuickTime 5
- Broadcast from a sequence grabber source
  - Audio devices: microphone, CD, or DV Audio source
  - Video devices: DV Camera, or USB camera (CritterCam)
- Broadcast can be unicast or multicast
- Broadcast is user-configurable using a dialog

Multicast Broadcast Configuration



Anatomy of a Presentation



Broadcasting Using the QTJava Presentation API

- Presentation API in quicktime.streaming package
- A streaming session is represented by a Presentation object
- Streaming session configurable via a settings dialog
- Streamlined API available now, additional features in a future release

Demo

Broadcasting Using the Presentation API

Michael Hopkins

Presentation API Details

- Creating a Presentation

```
Presentation.fromFile (
        mySDPfile,
        myPresentationParams);
```

- Showing the Settings Dialog

```
SettingsDialog dialog =
        new SettingsDialog(pres);
```

NOTE: The user is responsible to choose settings appropriate for the type of broadcast as specified by the SDP file

Presentation API Details (Cont.)

- Preparing and starting a Presentation

```
pres.preroll();
pres.start();
```

- Stopping a Presentation

```
pres.stop();
```

Just One More Thing…

- Setting up an idle routine

```
TaskThread tasker = new
    TaskThread (“Broadcaster Idler”, 20);
```

- Writing the execute method

```
public final void task ()
    throws QTException {
        pres.idle (null);
}
```

Transition Enhancements

Jon Summers
QTJava Engineering

Transitions Enhancements

- Filters and Transitions supported between any combination of still images and movie files
  - QTFilter applied to a single source
  - QTTransition applied between two sources

QTEffects Dialog Enhancements

- QTEffects dialog ParameterDialog
- showParameterDialog in QTJava 4

```
AtomContainer ParameterDialog.showParameterDialog (
    AtomContainer effectsList, int pdOptions);
```

- showParameterDialog extended in QTJava 5

```
AtomContainer  ParameterDialog.showParameterDialog (
    EffectsList effectsList, int pdOptions,
    AtomContainer parameters, String title, Pict[ ] pictArray);
```

  - Re-edit parameter container of current effect
  - Customize with optional preview pictures

QTJava 5 Enhancements

AtomContainer.insertChild

- Extended data types for inserting child atom data

  atomcontainer.insertChild(parent, atomType, id, index, data)

  – Simple data types are **byte, short, int and float**

  – Object data types are **byte[ ], QTByteObject, QTPointerRef and QTHandleRef**

- Tween type atom data is an array of start and end samples

  TweenData extends QTByteObject

  Overloaded constructor **new TweenData(TYPE start, TYPE end);**

  TYPE : short, int, Fixed, float, double, Point, FixedPoint, Matrix, Color or GraphicsModeWithColor

QTJava 5 New Features

- XML functionality
  - Support for XML Parser in QuickTime 5
  - XML data classes

    XMLParser, XMLDoc, XMLElement, XMLAttribute and XMLContent
  - Callbacks during parsing of XML

    StartDoc, EndDoc, StartElement, EndElement PreProcessor, CharData and Comment callbacks

New Features in QTJava 5: Drawing Notification Registration

Michael Hopkins
QTJava Engineering

New Features in QTJava 5—Drawing Notification

- Registration for Drawing Notification
- Notification when drawing and updating occurs
  - Compositor
  - Effects
  - MoviePresenter

New Features in QTJava 5—Drawing Notification Registration

- Provides registration services
- Listener implements DrawingListener interface:

```
public class MyListener implements DrawingListener {
   public void drawingComplete(QTDrawable drawable) {
         // your code goes here
} }
```

- Register your listener with the component:

```
MyListener listener = new MyListener();
theCompositor.addDrawingListener(listener);
```

QTJava 5 Additional New Features

- QTCanvas new sizing option
  kInitialSizeNoResize
- StdQTConstants5 interface
  - Represents newly defined constants in QT 5 API
  - StdQTConstants4 represents QT 4 API constants
  - StdQTConstants represents QT 3 and previous APIs

Who to Contact

General Developer Support

Public mailing list for QTJava Developers

lists.apple.com

For General Developer Information

QTJava SDK (Sample Code, Documentation)

developer.apple.com/quicktime/qtjava

For Seeding Enrollment

Must be Registered Apple Developer

qtjava@apple.com

Roadmap

FF109 QuickTime Feedback Forum: Let your voice be heard!	Room J1 **Thurs., 3:30pm**
508 Java Graphics: Learn about the new Java Graphics model	Civic **Thurs., 3:30pm**
208 Audio Services in Mac OS X: Hear about Core Audio Services	Room J2 **Thurs., 5:00pm**
509 Java Security: Learn how to sign your QTJava applet	Room J2 **Fri., 9:00am**
FF117 Java Feedback Forum: Bring your Java constructive criticism	Room J1 **Fri., 10:30am**

Q&A

Alan Samuel
Java Technology Manager
qtjava@apple.com

http://developer.apple.com/quicktime/qtjava

WWDC
Worldwide Developers Conference 2001